D2-RST: Dual-Dimensional Residual Side Tuning for Mitigating Feature Forgetting in Parameter-Efficient Transfer Learning

BMVC 2025 Submission # 212

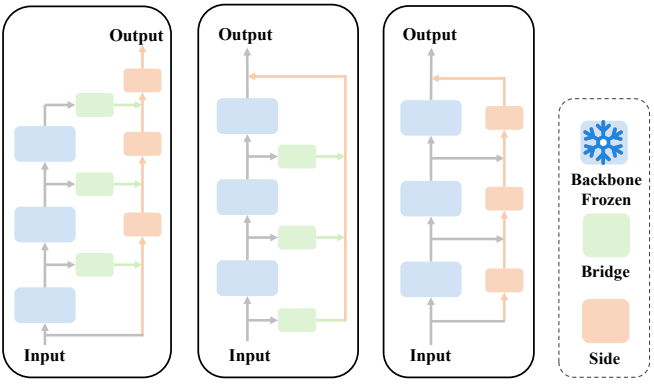
Abstract

Existing fine-tuning methods for pre-trained models, including parameter-efficient transfer learning (PETL) approaches, suffer from severe feature forgetting in deep layers due to progressive spectral decay. To address this issue, we present **Dual-Dimensional Residual Side Tuning (D2-RST)**, a novel PETL framework designed to mitigate feature forgetting by jointly optimizing aggregated features, i.e., residuals, across both embedding and spatial dimensions. Specifically, D2-RST employs a dual-block side tuning structure: Collect Blocks extract inter-layer information into residuals while Feed Blocks strategically reintegrate them back into the backbone. This parallel processing framework with low-rank linear mappings applied to residuals effectively stabilizes low-frequency components while reducing memory cost. Additionally, D2-RST introduces a spatial-dimension pathway in parallel with the conventional feature-dimension pathway, followed by cross-dimensional fusion via learnable scalars at each Feed Block, thereby effectively suppressing low-frequency attenuation in deeper layers. To further reduce redundancy, we propose a parameter-sharing strategy for LoRA matrices within Collect Blocks, where low-rank projections are shared across multiple layers. Extensive experiments on several benchmarks demonstrate that D2-RST not only outperforms existing PETL methods in accuracy but also significantly reduces parameter overhead by explicitly suppressing deep-layer feature forgetting.

1 Introduction

The paradigm of large-scale pre-training followed by fine-tuning has become the cornerstone of modern machine learning, driving significant advancements across various domains such as natural language processing, computer vision , and beyond. As the scale of these pre-trained models continues to expand, fine-tuning the entire parameter set has become increasingly impractical due to prohibitive computational and memory demands. This challenge is particularly pronounced in scenarios with limited computational resources or when deploying models on edge devices.

In response to these challenges, Parameter-Efficient Transfer Learning (PETL) methods have emerged as a promising solution. PETL approaches aim to adapt large pre-trained models to new tasks by updating only a small subset of parameters, thereby reducing the computational overhead and minimizing the risk of overfitting. To mitigate the memory challenges inherent in fine-tuning large pre-trained models, side tuning strategies have been



(a) LST (b) DTL (c) D2-RST / RST

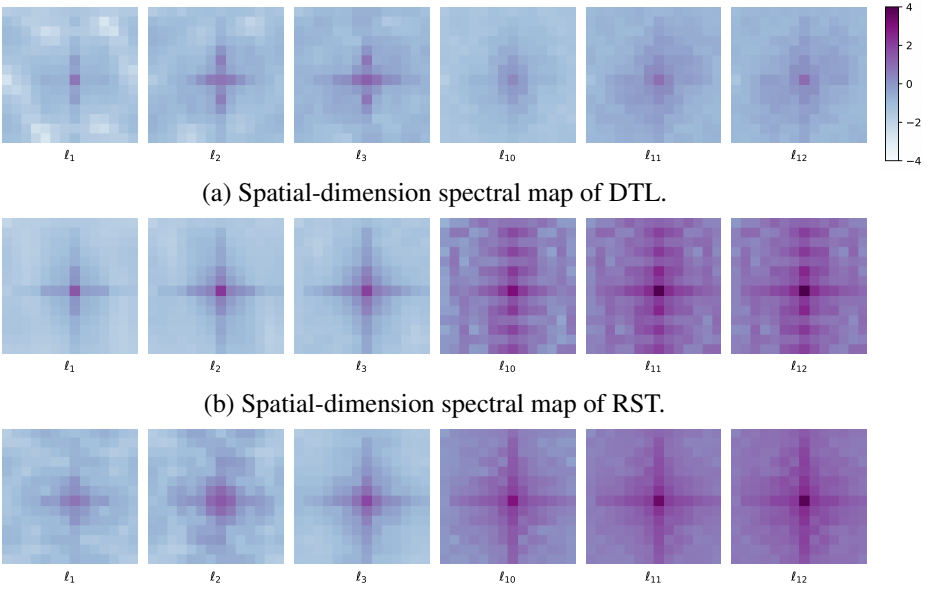
Figure 1: Comparative Architectures of Side Tuning Methods. In all subfigures, blue, green, orange elements respectively represent the frozen backbone network, bridge blocks that directly process backbone information, and side paths responsible for processing aggregated features. (a) LST: Combines bridge blocks and side paths. (b) DTL: Utilizes only bridge blocks. (c) D2-RST / RST: Employs only side paths, focusing on aggregated feature processing.

proposed. These strategies decouple the trainable modules from the backbone network by introducing parallel side networks or lightweight modules, thereby effectively reducing GPU memory usage. By eliminating the need to store extensive intermediate gradients within the backbone network, side tuning not only maintains parameter efficiency but also enhances the feasibility of fine-tuning large-scale models in resource-limited settings.

Although side tuning effectively addresses memory consumption by decoupling the trainable components, existing side tuning methods often suffer from severe feature forgetting in deeper layers due to progressive spectral decay and unstable low-frequency energy ratios across layers. These limitations result in suboptimal preservation of both global structural patterns (low-frequency components) and local discriminative details (high-frequency components), which can impede the overall performance and adaptability of fine-tuned models, especially in tasks requiring nuanced feature representations or operating under low-data regimes. We compare the architectures of widely used side tuning methods in Fig. 1.

To address these fundamental drawbacks, we first introduce **Residual Side Tuning (RST)**, a novel PETL framework that stabilizes low-frequency components through low-rank linear mappings applied to aggregated features (residuals). RST employs a dual-block architecture: Collect Blocks extract inter-layer information into residuals while Feed Blocks reintegrate them back into the backbone. By focusing on residuals rather than raw backbone features, RST explicitly suppresses spectral degradation and mitigates deep-layer feature forgetting.

Building upon RST, we further propose **Dual-Dimensional Residual Side Tuning (D2-RST)** to explicitly reduce low-frequency attenuation. D2-RST introduces a spatial-dimension pathway in parallel with the conventional feature-dimension pathway at each Feed Block, followed by cross-dimensional fusion via learnable scalars. This dual-path design decouples spectral dynamics across embedding and spatial domains, further suppressing low-frequency attenuation in deeper layers, as shown in Fig. 2 and Fig. 3 (a). The forgetting phenomenon can also be represented by computing layer-wise correlation between aggregated features and backbone features, as presented by Fig. 3 (b)&(c). Additionally, we implement a parameter-



(a) Spatial-dimension spectral map of DTL.

(b) Spatial-dimension spectral map of RST.

(c) Spatial-dimension spectral map of D2-RST.

Figure 2: Spatial-dimension spectral map of various architectures. (a) DTL. (b) RST. (c) D2-RST. DTL has the lowest low-frequency magnitude, while RST enhances the low-frequency magnitude. D2-RST further focuses on low-frequency component and suppresses high-frequency noises.

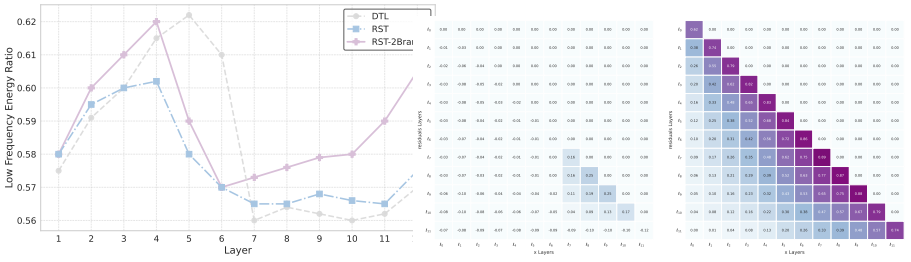
sharing strategy for LoRA matrices within Collect Blocks, enabling efficient utilization of model parameters through shared adaptations across multiple layers while preserving the diversity of extracted features. Our contributions can be summarized as follows:

1. We identify that the quick forgetting of low-frequency information is the bottleneck of existing side tuning methods.

2. We propose RST, a novel PETL framework that employs low-rank linear mappings on aggregated features within a dual-block architecture to significantly stabilize low-frequency components.

3. We further develop D2-RST by introducing a dual-dimensional adaptation mechanism in Feed Blocks, where spatial-dimension pathways are introduced to complement feature-dimension processing. By enforcing explicit modeling of spatial patterns and cross-dimensional feature fusion, D2-RST enhances low-frequency feature magnitude and energy through cross-dimensional interactions.

To comprehensively evaluate the effectiveness of D2-RST, we conduct extensive experiments across multiple benchmarks, including VTAB-1K [30], VTAB-100 built on VTAB-1K, few-shot learning, and domain generalization. Our experiments demonstrate that D2-RST consistently outperforms existing PETL methods in accuracy, particularly in low-shot learning scenarios. Additionally, it exhibits favorable scaling properties as model size increases. To further validate the strengths, we perform ablation studies that confirm the contributions of its key components.



(a) Frequency ratio of various models. (b) Feature similarity of DTL and (D2-)RST.

Figure 3: (a) The frequency ratio of various models. (b) The layer-wise correlation analysis measuring similarity (cosine similarity) between aggregated features and corresponding backbone layer inputs across all preceding blocks, where (D2-)RST (right) demonstrates stronger inter-layer correlations compared to DTL (left), indicating less information forgetting.

2 Method

We introduce Dual-Dimensional Residual Side Tuning (D2-RST), a novel parameter-efficient transfer learning framework, as shown in Fig. 4. Section 2.1 details its dual-block structural design (also the structure of RST). We then present the dual-dimensional design in Section 2.2. Finally, we describe our parameter sharing strategy within Collect Blocks in Section 2.3.

2.1 Dual-Block Architecture with Low-Rank Mapping for Residuals

2.1.1 Dual-Block Framework

(D2-)RST employs a dual-block framework comprising Collect Blocks and Feed Blocks, which operate in parallel to specific sections of the frozen backbone network, thereby preserving its pre-trained knowledge. Only the Collect and Feed Blocks are learnable and updated during training. The architecture of D2-RST is shown in Fig. 4, where the only difference between D2-RST and RST is that RST only contains the upper pathway of Feed blocks.

Collect Blocks are aligned with the first six blocks of the Vision Transformer (ViT) backbone, while Feed Blocks correspond to the last six blocks. These blocks extract inter-layer residuals that capture task-specific features through low-rank linear mappings, efficiently aggregating side information. During forward propagation, Collect Blocks gather residuals from the initial backbone layers, which Feed Blocks then reintegrate back into the backbone. This reintegration allows the backbone to adapt its feature representations based on the aggregated and refined task-specific information from the side path.

In backward propagation, gradients flow exclusively through the side path and the last six backbone blocks parallel to the Feed Blocks, limiting gradient backpropagation to the middle of the backbone and thus reducing memory usage. Unlike Ladder Side-Tuning (LST), which does not reintegrate information before the output layer, (D2-)RST enables gradient flow through the latter backbone layers. This design achieves a balanced trade-off between performance and memory efficiency, enhancing the model's ability to adapt to new tasks while maintaining lower memory consumption and preserving the integrity of the backbone's pre-trained knowledge.

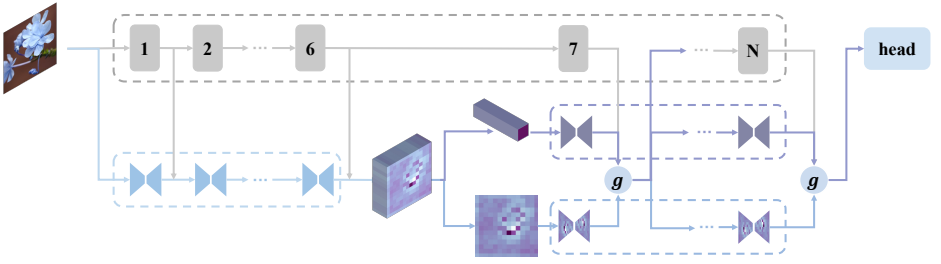


Figure 4: The architecture of D2-RST. The backbone blocks are frozen (shaded in gray). As for the side tuning path, the first six blocks are Collect Blocks (shaded in light blue), responsible for aggregating features from the backbone network. The sequential blocks, referred to as Feed Blocks, in which the upper and lower pathways respectively process the feature dimension and spatial dimension of the image. The outputs from the two pathways are aggregated with their contribution adjusted by a gating function g after each block. The aggregated information is then fed back to the backbone. The Feed Blocks of RST only have the upper Blocks, with the rest of the architecture remaining the same as in D2-RST.

2.1.2 Low-Rank Linear Mapping for Residuals

To alleviate the forgetting of useful low-frequency information in deeper layers, we implement low-rank linear mapping on the residuals extracted by both Collect Blocks and Feed Blocks, instead of on the inputs of backbone blocks. To be mentioned, the item "residuals" refers to aggregated features. We propose two propositions as follows, and theoretically prove them in Appendix B. To be mentioned, we only derive the case of RST, and the conclusion is also applicable to D2-RST due to their similar structure.

Proposition 1 (Feature Aggregation Dynamics)**.** *For ViT blocks with feature dimension m and low-rank adaptation rank r, let $A^{(i)} \in \mathbb{R}^{m\times r}$ and $B^{(i)} \in \mathbb{R}^{r\times m}$ denote the LoRA matrices at layer i. The aggregated features under DTL and RST architectures respectively, satisfy:*

DTL:

$$s_1^{(i)}|_k = B^{(k)\top} A^{(k)\top} x^{(k)} \tag{1}$$

exhibiting uniform attention over historical features.

RST:

$$s_2^{(i)}|_k = B^{(i)\top} D_k A^{(k+1)\top} \cdot B^{(k)\top} A^{(k)\top} x^{(k)} \tag{2}$$

establishing layer-adaptive feature composition through matrix chain multiplication.

Proof. Please refer to Appendix B.1. □

Proposition 2 (Gradient Sensitivity Characterization)**.** *The sensitivity of aggregated features to backbone activations reveals fundamental architectural differences:*

DTL:

$$\frac{\partial s_1^{(i)}}{\partial x^{(k)\top}}|_l = B^{(l)\top} A^{(l)\top} \frac{\partial x^{(l)}}{\partial x^{(k)\top}} \tag{3}$$

RST:

$$\frac{\partial s_2^{(i)}}{\partial x^{(k)\top}}|_l = \prod_{j=0}^{i-l} B^{(i-j)\top} A^{(i-j)\top} \frac{\partial x^{(l)}}{\partial x^{(k)\top}}$$
$$= B^{(i)\top} D_l A^{(l+1)\top} B^{(l)\top} A^{(l)\top} \frac{\partial x^{(l)}}{\partial x^{(k)\top}} \quad (4)$$

where $D_l \in \mathbb{R}^{r\times r}$ is an implicit scaling matrix. The additional term $B^{(i)\top} D_l A^{(l+1)\top}$ in RST enables adaptive noise filtering through layer-wise decoding.

Proof. Please refer to Appendix B.2. □

Comparing Eq. (1) and Eq. (2), $s_2^{(k)}$ has an additional item $B^{(i)T} D_k A^{(k+1)T}$, which implies that the information extracted from the preceding layers will be decoded by the decoder of the current layer. This reveals that applying a low-rank linear mapping to the aggregated information can make the features of the historical layers more compatible with the current features, thereby enhance the model's overall information extraction capabilities.

Comparing Eq. (3) and Eq. (4), $s_2^{(k)}$ has an additional item $B^{(i)T} D_l A^{(k+1)T}$, which implies that the sensitivity will be decoded by the decoder of the current layer. This reveals that our architecture can reduce the sensitivity to noise and irrelevant information introduced by the backbone network by filtering and refining residuals before reintegration, thus exhibiting higher ratio of low-frequency energy as illustrated in Fig. 2 and Fig. 3.

2.2 Dual-Dimensional Design for Explicit Spatial Modeling

To further enhance the capability of D2-RST in mitigating feature forgetting, we introduce a dual-dimensional design that explicitly models the spatial dimension of image features. As illustrated in Fig. 4, the dual-dimensional design integrates a parallel pathway dedicated to processing the spatial dimension alongside the conventional feature dimension. This dual-path structure allows for simultaneous handling of both embedding and spatial dimensions, thereby enriching the aggregated feature representations.

In the dual-dimensional framework, each Feed Block comprises two distinct pathways: an upper pathway that processes the feature dimension and a lower pathway dedicated to the spatial dimension. After each processing step, the outputs from these two pathways are aggregated using a gating function g, implemented as a learnable scalar. This gating mechanism dynamically adjusts the contribution of each pathway, ensuring a balanced integration of feature and spatial information. The aggregated output is then reintegrated into the backbone network, allowing the model to refine its feature representations based on the combined spatial and feature dimension information.

As shown in Fig. 2 and Fig. 3, by explicitly modeling the spatial dimension, the dual-dimensional design not only stabilizes the low-frequency components but also fosters a more robust interaction between spatial and feature dimensions. This interaction further suppresses the attenuation of low-frequency energy in the deeper layers, leading to more stable and enriched feature representations.

2.3 Encoder Parameter Sharing Strategy

To further reduce redundancy and optimize parameter efficiency, (D2-)RST employs a parameter-sharing strategy within the Collect Blocks. Specifically, we share the LoRA-A matrices across multiple Collect Blocks, allowing the low-rank projections to be reused across different layers. This strategy decreases the number of trainable parameters by 21% without compromising the model's ability to capture and aggregate task-specific features. By sharing these adapters, we maintain the flexibility and expressiveness of individual blocks while achieving substantial reductions in overall parameter overhead.

To identify the effect of parameter sharing, we conduct t-SNE on the aggregated features of RST with or without parameter sharing and visualize the feature richness of them in Section E.4. As shown in Fig. 6, parameter sharing does not compromise the feature richness of learned features.

The conclusions derived in Section 2.1 remain applicable and are even presented in a clearer form, as proved in Appendix. C.

Based on the methodologies, we construct four variants of the (Dual-Dimensional) Residual Side Tuning framework: RST, RST-NS, D2-RST, D2-RST-NS, where the postfix "NS" indicates non-shared parameter of encoders.

3 Experiments

To comprehensively evaluate the effectiveness of the proposed models, we conduct extensive experiments across multiple benchmarks, including few-shot learning in Section 3.1, VTAB-1K [30] in Section 3.2, and domain generalization in Section 3.3. And the detailed experimental settings are exhibited in Appendix D.

Besides, we conduct extensive ablation studies to verify the properties of RST applied in Appendix E. Specifically, in Appendix E.1, we perform an ablation study on the impact of the low-rank linear mapping by comparing RST-NS with DTL. In Appendix E.2, we further investigate the effectiveness of the dual-dimensional processing on the VTAB-1K benchmark. Finally, in Appendix E.3, we conduct an ablation study on the effect of parameter sharing, also evaluated on the VTAB-1K benchmark.

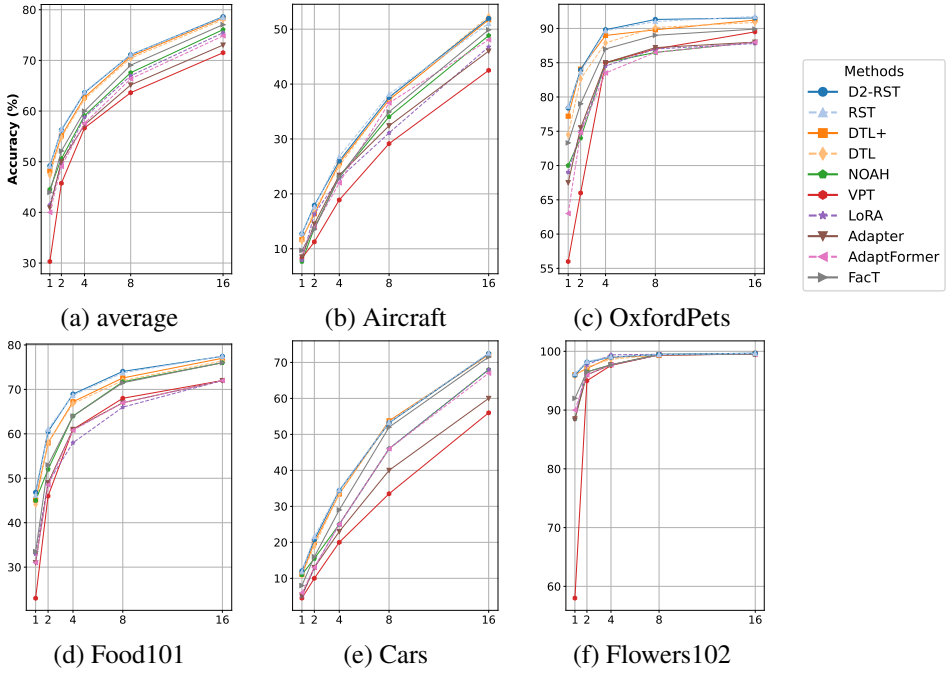
3.1 Experiments on Few-Shot Learning

To evaluate the few-shot learning capabilities of RST, we conduct experiments on five fine-grained benchmarks according to the settings in Appendix D.

As illustrated in Fig. 5, the proposed RST and D2-RST models consistently outperform all baseline PETL methods across various few-shot scenarios. Furthermore, we observe that the proposed models exhibit average improvements of over 1% compared to the previous best method DTL+. These comparisons underscore the exceptional performance of the RST variants in few-shot learning tasks, validating the effectiveness of our proposed approach.

3.2 Experiments on VTAB-1K

We conduct image classification tasks on VTAB-1K benchmark based on the setting illustrated in Appendix D.



(a) average (b) Aircraft (c) OxfordPets

(d) Food101 (e) Cars (f) Flowers102

Figure 5: Top-1 accuracy on fine-grained few-shot benchmark with ViT-B/16 as the backbone. Note that our approach outperforms all baseline methods.

As shown in Table. 1, RST and D2-RST demonstrate both competitive and enhanced performance compared to previous work. Specifically, D2-RST outperforms all previous PEFT methods with fewer parameters than them. And RST attains an average accuracy of 76.7%, matching the performance of DTL while being more parameter-efficient. We also present the results of RST*, which shares the same architecture and parameter count with RST but consists entirely of Feed Blocks. RST* achieves a significant performance improvement over previous methods, with an increase of 1%.

3.3 Experiments on Domain Generalization

To assess the robustness of D2-RST and RST under domain shifts, we conduct domain generalization experiments according to Appendix D.

The results of the domain adaptation experiments are presented in Table. 2. We observe that, compared to the previous state-of-the-art methods, both D2-RST and RST achieve impressive gains in evaluation accuracy across all target domains, with average improvements reaching up to approximately 1.4%. These comparisons highlight the exceptional robustness of the them in addressing domain shift challenges and effectively demonstrate the superiority of the proposed method.

Table 1: Per-task fine-tuning results on VTAB-1k benchmark. The backbone is ViT-B/16, and we ignore the linear layer when calculating the number of learnable parameters. RST* has the same structure as RST but sets all blocks as Feed blocks.

		Natural							Specialized				Structured								
	#Params (M)	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	SUN397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Average
Traditional methods																					
Full	85.8	68.9	87.7	64.3	97.2	86.9	87.4	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	68.9
Linear	0	63.4	85.0	63.2	97.0	86.3	36.6	51.0	78.5	87.5	68.5	74.0	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	57.6
PETL methods																					
VPT-deep	0.60	**78.8**	90.8	65.8	98.0	88.3	78.1	49.6	81.8	96.1	83.4	68.4	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	72.0
BitFit	0.10	72.8	87.0	59.2	97.5	85.3	59.9	51.4	78.7	91.6	72.9	69.8	61.5	55.6	32.4	55.9	66.6	40.0	15.7	25.1	65.2
Adapter	0.16	69.2	90.1	68.0	98.8	89.9	82.8	54.3	84.0	94.9	81.9	75.5	80.9	65.3	48.6	78.3	74.8	48.5	29.9	41.6	73.9
LoRA	0.25	67.1	91.4	69.4	98.8	90.4	85.3	54.0	84.9	95.3	84.4	73.6	82.9	**69.2**	49.8	78.5	75.7	47.1	31.0	44.0	74.5
AdaptFormer	0.16	70.8	91.2	70.5	99.1	90.9	86.6	54.8	83.0	95.8	84.4	**76.3**	81.9	64.3	49.3	80.3	76.3	45.7	31.7	41.1	74.7
Compacter	0.15	71.9	89.0	69.7	99.1	90.7	82.7	56.1	86.0	93.5	82.4	75.3	80.2	63.4	47.4	77.2	78.1	53.5	27.3	39.8	74.2
SSF	0.21	69.0	92.6	**75.1**	99.4	91.8	90.2	52.9	**87.4**	95.9	87.4	75.5	75.9	62.3	**53.3**	80.6	77.3	54.9	29.5	37.9	75.7
NOAH	0.39	69.6	92.7	70.2	99.1	90.4	86.1	53.7	84.4	95.4	83.9	75.8	82.8	68.9	49.9	81.7	81.8	48.3	32.8	44.2	75.5
Convpass	0.33	72.3	91.2	72.2	99.2	90.9	**91.3**	54.9	84.2	96.1	85.3	75.6	82.3	67.9	51.3	80.0	85.9	53.1	**36.4**	44.4	76.6
FacT-TK	0.07	70.6	90.6	70.8	99.1	90.7	88.6	54.1	84.8	**96.2**	84.5	75.7	82.6	68.2	49.8	80.7	80.8	47.4	33.2	43.0	75.6
LST	2.38	59.5	91.5	69.0	99.2	89.9	79.5	54.6	86.9	95.9	85.3	74.1	81.8	61.8	52.2	81.0	71.7	49.5	33.7	45.2	74.3
DTL	0.04	69.6	**94.8**	71.3	99.3	91.3	83.3	56.2	87.1	**96.2**	86.1	75.0	82.8	64.2	48.8	81.9	**93.9**	53.9	34.2	**47.1**	76.7
Proposed methods																					
D2-RST	0.034	73.7	94.5	72.4	99.4	**91.9**	81.1	**57.4**	86.8	95.9	86.8	75.8	82.9	65.2	51.2	80.3	92.5	54.2	34.6	45.4	77.0
RST	**0.029**	73.7	94.1	71.8	**99.5**	91.6	82.1	**57.8**	86.8	95.9	86.6	75.0	82.7	64.3	50.7	81.5	87.6	**55.7**	32.1	46.1	76.7
RST*	**0.029**	76.9	94.0	73.1	**99.5**	91.5	88.9	57.2	87.3	**96.2**	**87.5**	74.6	**83.2**	64.9	51.6	**83.0**	88.2	54.6	33.5	**49.2**	**77.6**

Table 2: Top-1 accuracy on domain generalization experiments with ViT-B/16 as the backbone. Our method shows significant gains w.r.t baseline methods.

Method	Source	Target				
	ImageNet	-Sketch	-V2	-A	-R	Avg
Adapter	70.5	16.4	59.1	5.5	22.1	25.8
VPT	70.5	18.3	58.0	4.6	23.2	26.0
LoRA	70.8	20.0	59.3	6.9	23.3	27.4
NOAH	71.5	24.8	66.1	11.9	28.5	32.8
DTL	78.3	35.4	67.8	14.0	34.4	37.9
DTL+	**78.7**	35.7	67.8	14.2	34.4	38.0
D2-RST	77.0	36.5	**68.6**	15.4	**36.9**	**39.4**
RST	77.2	**36.8**	68.4	**15.6**	36.4	39.3

4 Conclusion

We introduce **Dual-Dimensional Residual Side Tuning (D2-RST)**, a novel parameter-efficient transfer learning framework designed to mitigate feature forgetting and progressive spectral decay in deep layers by utilizing a dual-block architecture with low-rank linear mapping applied to aggregated features, as well as facilitating explicit dual-dimensional modeling and cross-dimensional interaction. Extensive experiments are performed across multiple benchmarks, including VTAB-1K, few-shot learning, and domain generalization. These experiments demonstrate that D2-RST consistently outperforms existing PETL methods in accuracy. Additionally, it exhibits favorable properties of low-frequency energy stabilization. To further validate the strengths of D2-RST, we perform ablation studies that confirm the contributions of its key components.

References

[1] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597 – 1607, 2020.

[4] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision Transformer Adapter for Dense Predictions. *arXiv e-prints*, art. arXiv:2205.08534, May 2022. doi: 10.48550/arXiv.2205.08534.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:52967399.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv e-prints*, art. arXiv:2010.11929, October 2020. doi: 10.48550/arXiv.2010.11929.

[8] Minghao Fu, Ke Zhu, and Jianxin Wu. Dtl: Disentangled transfer learning for visual recognition, 2024. URL https://arxiv.org/abs/2312.07856.

[9] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural Adversarial Examples. *arXiv e-prints*, art. arXiv:1907.07174, July 2019. doi: 10.48550/arXiv.1907.07174.

[10] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329, 2021. doi: 10.1109/ICCV48922.2021.00823.

[11] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. *arXiv e-prints*, art. arXiv:1902.00751, February 2019. doi: 10.48550/arXiv.1902.00751.

[12] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://aclanthology.org/P18-1031/.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv e-prints*, art. arXiv:2106.09685, June 2021. doi: 10.48550/arXiv.2106.09685.

[14] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning. *arXiv e-prints*, art. arXiv:2203.12119, March 2022. doi: 10.48550/arXiv.2203.12119.

[15] Shibo Jie and Zhi-Hong Deng. Convolutional Bypasses Are Better Vision Transformer Adapters. *arXiv e-prints*, art. arXiv:2207.07039, July 2022. doi: 10.48550/arXiv.2207.07039.

[16] Shibo Jie and Zhi-Hong Deng. Fact: factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i1.25187. URL https://doi.org/10.1609/aaai.v37i1.25187.

[17] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. doi: 10.1109/ICCVW.2013.77.

[18] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: a new baseline for efficient model tuning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[19] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv e-prints*, art. arXiv:1711.05101, November 2017. doi: 10.48550/arXiv.1711.05101.

[20] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-Grained Visual Classification of Aircraft. *arXiv e-prints*, art. arXiv:1306.5151, June 2013. doi: 10.48550/arXiv.1306.5151.

[21] Otniel-Bogdan Mercea, Alexey Gritsenko, Cordelia Schmid, and Anurag Arnab. Time-, Memory- and Parameter-Efficient Visual Adaptation. *arXiv e-prints*, art. arXiv:2402.02887, February 2024. doi: 10.48550/arXiv.2402.02887.

[22] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

[23] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

[24] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet? *arXiv e-prints*, art. arXiv:1902.10811, February 2019. doi: 10.48550/arXiv.1902.10811.

[25] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning, 2022. URL https://arxiv.org/abs/2206.06522.

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

[27] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3eefceb8087e964f89c2d59e8a249915-Paper.pdf.

[28] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

[29] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *arXiv e-prints*, art. arXiv:1905.04899, May 2019. doi: 10.48550/arXiv.1905.04899.

[30] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark. *arXiv e-prints*, art. arXiv:1910.04867, October 2019. doi: 10.48550/arXiv.1910.04867.

[31] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. *arXiv e-prints*, art. arXiv:1710.09412, October 2017. doi: 10.48550/arXiv.1710.09412.

[32] Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In *Computer*

Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III, page 698–714, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58579-2. doi: 10.1007/978-3-030-58580-8_41. URL https://doi.org/10.1007/978-3-030-58580-8_41.

[33] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural Prompt Search. *arXiv e-prints*, art. arXiv:2206.04673, June 2022. doi: 10.48550/arXiv.2206.04673.

A Related Work

Challenges in Fine-Tuning Large Pre-Trained Models Large pre-trained models have significantly advanced fields such as natural language processing (NLP), computer vision (CV), and vision-language (VL) tasks by leveraging vast datasets to develop comprehensive and generalizable representations. However, fine-tuning [6, 12] these massive models for specific downstream tasks is computationally expensive and memory-intensive. Additionally, fully fine-tuning all parameters can lead to catastrophic forgetting, where the model loses its pre-trained knowledge when adapting to new tasks. Traditional fine-tuning approaches like linear probing [3, 28] , which involve training only a linear classifier on frozen features, often underperform compared to full fine-tuning, highlighting the need for methods that balance parameter efficiency and training resource requirements.

Parameter-Efficient Transfer Learning (PETL) Methods Recent advances in parameter-efficient transfer learning have produced diverse adaptation strategies. Adapters [4, 11] introduce trainable bottleneck layers between transformer blocks for task-specific feature transformation, while LoRA [13] achieves parameter reduction through low-rank decomposition of weight update matrices. BitFit [1] demonstrates surprising effectiveness by selectively updating bias terms, establishing a minimalistic tuning paradigm. In vision domains, VPT [14] pioneers learnable prompt injection at transformer inputs, whereas SSF [18] enables feature adaptation through element-wise scaling and shifting operations. Fact [16] enhances low-rank tuning efficiency via tensor decomposition techniques, and ConvPass [15] incorporates convolutional layers for localized spatial adaptation. NOAH [33] further advances the field by automating architecture selection across multiple PETL components through neural architecture search.

Side-Tuning Methods Side Tuning [32] enhances pre-trained backbone networks by integrating auxiliary side networks without modifying the backbone's original parameters, reducing fine-tuning memory overhead and enabling efficient knowledge transfer. Ladder Side-Tuning (LST) [25] separates trainable parameters from the backbone with a lightweight side network, effectively reducing memory consumption but potentially degrading performance on challenging tasks. LoSA [21] utilizes low-rank side adaptors and prevents gradient flow within the backbone, resulting in a substantial reduction in memory usage. Disentangled Transfer Learning (DTL) [8] builds on LST by introducing a Compact Side Network (CSN) with low-rank linear mappings, reducing memory footprint and improving performance on difficult tasks. Fig. 1 shows the main structure of them.

These existing side-tuning-based PETL methods demonstrate the potential for enhancing backbone networks efficiently. However, they often struggle with preserving low-frequency information. To address this shortcoming, we propose a novel RST and D2-RST, enhancing low-frequency feature magnitude and energy while maintaining parameter efficiency.

B Proofs and Derivations

B.1 Proof of Proposition 1

For the structure of DTL that the inputs of each ViT block are processed by the low-rank linear mapping, the recursive formula for the aggregate features s is:

$$s^{(i)} = s^{(i-1)} + B^{(i)^T} A^{(i)^T} x^{(i)}$$

where $x^{(i)} = f^{(i)}(x^{(i-1)})$.

Obviously, we get:

$$s^{(1)} = B^{(1)^T} A^{(1)^T} x^{(1)}$$

Thus we can get the expression of aggregated features, as shown in Eq. (5):

$$s_1^{(i)} = \sum_{j=1}^{i} B^{(j)^T} A^{(j)^T} x^{(j)} \tag{5}$$

The observation indicates that DTL exhibits an equal level of attention to information across all previous layers.

For the structure of RST that the aggregated features parallel to each ViT block are processed by the low-rank linear mapping, the recursive formula for the aggregate information s is:

$$s^{(i)} = B^{(i)^T} A^{(i)^T} s^{(i)} + x^{(i+1)}$$

where $x^{(i+1)} = f^{(i+1)}(x^i)$.

Obviously, we get:

$$s^{(1)} = B^{(1)^T} A^{(1)^T} x^{(1)} + x^{(2)}$$

Thus we can get the expression of aggregated features, as shown in Eq. (6):

$$s_2^{(i)} = \sum_{k=1}^{i} \prod_{j=0}^{i-k} B^{(i-j)T} A^{(i-j)T} x^{(k)} + x^{(i+1)} \tag{6}$$

For a fixed k, the aggregated feature of $s_1^{(i)}|_k$ and $s_2^{(i)}|_k$ related to x_k can be expressed in Eq. (1) and Eq. (2), respectively.

$$\begin{aligned} s_1^{(i)}|_k &= B^{(k)T} A^{(k)T} x^{(k)} \\ s_2^{(i)}|_k &= \prod_{j=0}^{i-k} B^{(i-j)T} A^{(i-j)T} x^{(k)} \\ &= B^{(i)T} A^{(i)T} \cdots B^{(k+1)T} A^{(k+1)T} B^{(k)T} A^{(k)T} x^{(k)} \\ &= B^{(i)T} D_k A^{(k+1)T} \cdot B^{(k)T} A^{(k)T} x^{(k)} \end{aligned}$$

where $D_k \in \mathbb{R}^{r \times r}$, indicating a scaling matrix.

B.2 Proof of Proposition 2

Similarly, we can get the sensitivity of the aggregated features to the backbone information of the two structures, as shown in Eq. (7) and Eq. (8):

$$\frac{\partial s_1^{(i)}}{\partial x^{(k)T}} = \sum_{l=k}^{i} B^{(l)^T} A^{(l)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}} \tag{7}$$

$$\frac{\partial s_2^{(i)}}{\partial x^{(k)T}} = \sum_{l=k}^{i} (\prod_{j=0}^{i-l} B^{(i-j)^T} A^{(i-j)^T}) \frac{\partial x^{(l)}}{\partial x^{(k)T}} + \frac{\partial x^{(i+1)}}{\partial x^{(k)T}} \tag{8}$$

For a fixed l, the sensitivity of $s_1^{(i)}$ and $s_2^{(i)}$ to the backbone information of previous layers can be expressed as:

$$\frac{\partial s_1^{(i)}}{\partial x^{(k)T}}|_l = B^{(l)^T} A^{(l)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}}$$

$$\frac{\partial s_2^{(i)}}{\partial x^{(k)T}}|_l = \prod_{j=0}^{i-l} B^{(i-j)^T} A^{(i-j)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}} = B^{(i)^T} D_l A^{(l+1)T} B^{(l)^T} A^{(l)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}}$$

C Proof of The Applicability of Conclusions in the Secenario of Parameter Sharing

Based on Eq. (1) and Eq. (2), we can get:

$$\begin{aligned} s_1^{(i)}|_k &= B^{(k)T} A^{(k)T} x^{(k)} = B^{(k)T} A x^{(k)} \\ s_2^{(i)}|_k &= B^{(i)T} D_k A^{(k+1)T} \cdot B^{(k)T} A^{(k)T} x^{(k)} \\ &= B^{(i)T} D'_k A x^{(k)} \end{aligned}$$

Moreover, the sensitivity deducted in Eq. (3) and Eq. (4):

$$\begin{aligned} \frac{\partial s_1^{(i)}}{\partial x^{(k)T}}|_l &= B^{(l)^T} A^{(l)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}} = B^{(l)^T} A \frac{\partial x^{(l)}}{\partial x^{(k)T}} \\ \frac{\partial s_2^{(i)}}{\partial x^{(k)T}}|_l &= B^{(i)^T} D_l A^{(l+1)T} B^{(l)^T} A^{(l)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}} \\ &= B^{(i)^T} D'_l A \frac{\partial x^{(l)}}{\partial x^{(k)T}} \end{aligned}$$

where $D'_k, D'_l \in \mathbb{R}^{r \times r}$, also indicating a scaling matrix.

Therefore, the conclusions we derived in Section 2.1 remain applicable and are even presented in a clearer form.

D Experimental Settings

This section outlines our experimental settings, including the selection of pre-trained backbones, baseline methods for comparison, and implementation details.

D.1 Pre-trained Backbone

For our experiments, we exclusively utilize the Vision Transformer Base/16 (ViT-B/16) [7] model, which consists of approximately 86 million parameters and is pre-trained on the ImageNet-21K dataset [5]. The ViT-B/16 backbone is chosen due to its strong scalability and adherence to the scaling laws, which facilitate efficient adaptation across various tasks. Its widespread adoption in prior works underscores its robustness and versatility, making it an ideal foundation for evaluating the performance and scalability of the RST framework.

The baseline methods are mentioned in Section A.

D.2 Implementation Details

We adhere to the implementation protocols established in prior works [8, 16, 18] to ensure consistency and reproducibility in our experiments. Specifically, we employ the AdamW optimizer [19] with a cosine learning rate schedule. All models are fine-tuned for 100 epochs with a batch size of 32. For (D2-)RST, the rank r of the low-rank linear mappings of all blocks is set to 2. We configure the number of Collect Blocks to 6, indicating that half of the later blocks' outputs are calibrated by integrating residual information. This configuration ensures a balance between adaptation capacity and parameter efficiency.

Unlike some previous methods [18, 33], we restrict our approach to standard data augmentation techniques and do not incorporate additional strategies such as mixup [31], cutmix [29], or label smoothing [26]. This decision streamlines the training process and highlights the intrinsic effectiveness of the RST framework. Comprehensive details of our training hyperparameters and configurations are provided in the supplementary material.

D.3 Benchmarks for Evaluation

We conduct the experiments following the setting of DTL, LoRA, NOAH, etc..

The VTAB-1K benchmark [30] is designed to evaluate the generalization ability of transfer learning approaches across diverse image domains. It comprises 19 distinct datasets categorized into three groups: 1) Natural images captured by standard cameras, including everyday objects and scenes, reflecting common visual recognition tasks. 2) Specialized images captured by specialist equipment, often involving medical imaging, satellite imagery, and other domains requiring specialized knowledge. 3) Structured images generated in simulated environments, including synthetic data for tasks like depth prediction and object counting. Each dataset contains exactly 1,000 training examples, making it a stringent test for Parameter-Efficient Transfer Learning (PETL) methods. The diversity of VTAB-1K spans various task-specific objectives, including classic visual recognition, object counting, and depth prediction, among others. This variety ensures that any proposed method must demonstrate robust adaptability across different visual tasks and domains.

The few-shot learning experiments are conducted on five fine-grained benchmarks: Aircraft [20], Pets [23], Food-101 [2], Cars [17], and Flowers102 [22]. The paper reports the average accuracy on the train sets over three random seeds.

The domain generalization experiments follow the setup of noah [53] and dtl [8]. The training set consists of samples from the original ImageNet-1K training set, with each class containing 16 training images. The model is evaluated on four distinct datasets: ImageNet-Sketch [77] composed of sketch images sharing the same label space with ImageNet-1K, ImageNet-V2 [74] collected from different sources compared with ImageNet-1K, ImageNet-A [9] consisting of adversarial examples, and ImageNet-R [10] containing various artistic renditions of ImageNet-1K. The paper reports the average accuracy on the train sets over three random seeds.

E Ablation Study

We conduct various ablation experiments in this section.

E.1 The Effect of Applying Low-Rank Linear Mapping to Residuals

Since the difference between DTL and RST-NS is that DTL applies a low-rank linear mapping on backbone features, while RST-NS applies it on aggregated features, we can identify the effect of mapping objects by directly comparing them. In this section, we analyze their performance in the few-shot scenario.

The classification accuracy is shown in Table. 3. It can be concluded that RST-NS surpasses DTL on the vast majority of tasks across all datasets, with only two exceptions. Notably, it achieves over 2% accuracy gains on several datasets.

Datasets	Aircraft					OxfordPets					Food101				
	1-shot	2-shot	4-shot	8-shot	16-shot	1-shot	2-shot	4-shot	8-shot	16-shot	1-shot	2-shot	4-shot	8-shot	16-shot
RST-NS	**12.65**	**17.38**	**27.29**	**37.93**	51.33	**78.59**	**83.87**	**89.77**	**91.05**	**91.46**	**46.47**	**60.64**	**67.81**	**73.84**	**77.20**
DTL	11.44	16.95	25.07	37.40	**52.31**	74.51	82.64	87.88	90.14	90.87	44.15	58.01	66.80	71.97	76.40
	Cars					Flowers					Average				
	1-shot	2-shot	4-shot	8-shot	16-shot	1-shot	2-shot	4-shot	8-shot	16-shot	1-shot	2-shot	4-shot	8-shot	16-shot
RST-NS	**11.63**	**20.84**	**34.49**	**53.51**	71.50	**95.97**	**98.23**	**99.15**	**99.47**	**99.65**	**49.06**	**56.19**	**63.70**	**71.16**	**78.23**
DTL	10.95	18.95	33.51	53.27	**72.35**	95.81	97.30	98.60	99.37	99.61	47.37	54.77	62.38	70.43	78.03

Table 3: The classification accuracy of RST-NS and DTL across multiple tasks and various few-shot datasets.

E.2 The Effect of Dual-Dimensional Structure

We compare RST-NS with D2-RST-NS, and RST with D2-RST, evaluated on VTAB-1K, to identify the effect of dual-dimensional structure.

The classification accuracy is exhibited in Table. 4, where a significant performance gap in a group is highlighted in bold. It can be concluded that dual-dimensional structure

Table 4: Per-task fine-tuning results on VTAB-1k benchmark. The backbone is ViT-B/16, and we ignore the linear layer when calculating the number of learnable parameters. Significant performance gap in a group is highlighted in bold.

		Natural							**Specialized**				**Structured**								
	#Params (M)	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	SUN397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Average
D2-RST	0.034	73.7	94.5	72.4	99.4	91.9	81.1	57.4	86.6	95.9	86.8	**75.8**	82.9	**65.2**	51.2	80.3	**92.5**	54.2	**34.6**	45.4	77.0
RST	0.029	73.7	94.1	71.8	99.5	91.6	**82.1**	57.8	86.8	95.9	86.6	75.0	82.7	64.3	50.7	**81.5**	87.6	**55.7**	32.1	46.1	76.7
D2-RST-NS	0.034	73.0	94.3	72.1	99.5	91.5	81.4	57.3	86.8	95.8	86.9	**75.5**	82.6	**65.7**	50.8	80.0	**94.2**	55.3	**35.6**	45.0	77.1
RST-NS	0.029	73.8	94.2	72.1	99.4	91.7	81.2	57.6	86.9	95.9	86.4	74.7	82.9	64.4	**51.9**	**82.6**	87.5	56.0	32.2	**46.7**	76.8

significantly improves model performance on some datasets, validating the efficiency of the design.

E.3 The Effect of Parameter Sharing Strategy

We compare D2-RST with D2-RST-NS, and RST with RST-NS, evaluated on VTAB-1K, to identify the effect of parameter sharing strategy.

The classification accuracy is exhibited in Table. 5, where a significant performance gap in a group is highlighted in bold.

Table 5: Per-task fine-tuning results on VTAB-1k benchmark. The backbone is ViT-B/16, and we ignore the linear layer when calculating the number of learnable parameters. The significant performance gap in a group is highlighted in bold.

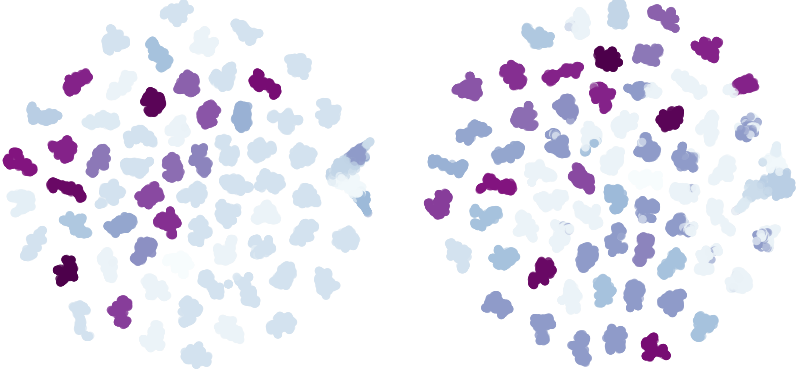
		Natural							**Specialized**				**Structured**								
	#Params (M)	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	SUN397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Average
D2-RST	0.034	73.7	94.5	72.4	99.4	91.9	81.1	57.4	86.6	95.9	86.8	75.8	82.9	65.2	51.2	80.3	92.5	54.2	34.6	45.4	77.0
D2-RST-NS	0.034	73.0	94.3	72.1	99.5	91.5	81.4	57.3	86.8	95.8	86.9	75.5	82.6	65.7	50.8	80.0	**94.2**	**55.3**	**35.6**	45.0	77.1
RST	0.029	73.7	94.1	71.8	99.5	91.6	**82.1**	57.8	86.8	95.9	86.6	75.0	82.7	64.3	50.7	81.5	87.6	55.7	32.1	46.1	76.7
RST-NS	0.029	73.8	94.2	72.1	99.4	91.7	81.2	57.6	86.9	95.9	86.4	74.7	82.9	64.4	**51.9**	**82.6**	87.5	56.0	32.2	46.7	76.8

It can be concluded that the parameter sharing strategy has no significant effect on model performance, thus it is feasible to apply this strategy.

E.4 The Effect of Parameter Sharing: A Feature Richness Perspective

To assess whether the parameter sharing strategy affects the richness of aggregated features, we conducted ablation experiments exclusively on D2-RST. Feature richness was evaluated using t-distributed Stochastic Neighbor Embedding (t-SNE) to visualize the separability of aggregated features. The degree of separability in the t-SNE plots serves as an indicator of feature richness, with higher separability implying more distinctive and informative feature representations.

Figure 6 presents t-SNE visualizations comparing D2-RST models with and without parameter sharing. The results reveal that the introduction of parameter sharing does not degrade the separability of features. For RST with or without parameter sharing, the results are similar. In fact, the feature distributions remain similarly distinct, indicating that feature richness is preserved despite parameter sharing. This finding aligns with our theoretical predictions, demonstrating that parameter sharing effectively reduces the number of trainable parameters without compromising the expressiveness or diversity of the extracted features. Consequently, the parameter sharing strategy employed in the (D2-)RST framework is both reasonable and effective, ensuring efficient parameter utilization while maintaining high performance.



(a) (b)

Figure 6: Feature Richness in D2-RST with/without parameter sharing. (a) Feature richness of D2-RST w/o parameter sharing. (b) Feature richness of D2-RST w/ parameter sharing. T-SNE visualizations on CIFAR100 of aggregated features indicating feature richness, where parameter sharing has little effect on feature richness.